and combination of databases, the scoring function **128** can represent any number and combination of scoring functions, and the search algorithm **122** can represent any number and combination of search algorithms.

[0024] FIG. **2** is a block diagram of another example of a system **200** for determining a recommended software-stack for a software item according to some aspects of the present disclosure. The system **200** includes a processor **202** communicatively coupled with a memory **204**. In some examples, the processor **202** and the memory **204** can be included in the same housing (e.g., in the server **112** of FIG. **1**) or remote from one another.

[0025] The processor **202** can include one processor or multiple processors. Examples of the processor **202** include a Field-Programmable Gate Array (FPGA), an application-specific integrated circuit (ASIC), a microprocessor, or any combination of these. The processor **202** can execute instructions **206** stored in the memory **204** to perform operations. In some examples, the instructions **206** can include processor-specific instructions generated by a compiler or an interpreter from code written in any suitable computer-programming language, such as C, C++, C#, etc.

[0026] The memory **204** can include one memory or multiple memories. In some examples, the memory **204** can be non-volatile such that it retains stored information when powered off. For instance, the memory **204** can include electrically erasable and programmable read-only memory (EEPROM), flash memory, or any other type of non-volatile memory. At least some of the memory **204** includes a non-transitory computer-readable medium from which the processor **202** can read instructions **206**. A non-transitory computer-readable medium can include electronic, optical, magnetic, or other storage devices capable of providing the processor **202** with computer-readable instructions or other program code. Examples of a non-transitory computer-readable medium can include magnetic disk(s), memory chip(s), ROM, random-access memory (RAM), an ASIC, a configured processor, optical storage, or any combination of these.

[0027] In some examples, the processor **202** can receive an input **110** specifying a target software item **208** and a characteristic of a computing environment **210** in which the target software item **208** is to be executed. The processor **202** can then generate software-stack candidates **124***a-n* for the target software item **208**. The software-stack candidates **124***a-n* can have unique configurations of software components. The processor **202** can also determine a respective score for each software-stack candidate of the software-stack candidates **124***a-n* based on the characteristic of the computing environment **210** and a unique configuration of software components forming the software-stack candidate. The processor **202** can calculate the respective score for each software-stack candidate as part of, or after, a search process for determining the software-stack candidates **124***a-n*. The processor **202** can select a particular software-stack candidate **124***n* from the software-stack candidates **124***a-n* as a recommended software-stack **114**. The processor **202** can make this selection based on the respective score **126***n* for the particular software-stack candidate **124***n* having a predefined attribute, such as the respective score **126***n* being the maximum score or the minimum score among all of the scores **126***a-n*. The processor **202** can then generate an output **120** indicating the recommended software-stack **114**, for example, to enable the recommended software-stack **114** to be included in the computing environment.

[0028] In some examples, the processor **202** can implement some or all of the steps shown in FIG. **3**. Other examples can include more steps, fewer steps, different steps, or a different order of the steps than is shown in FIG. **3**. Some or all of the steps may be repeated. The steps of FIG. **3** are discussed below with reference to the components discussed above from FIG. **2**.

[0029] In block **302**, the processor **202** receives an input **110** specifying a target software item **208** and a characteristic of a computing environment **210** in which the target software item **208** is to be executed. The processor **202** can receive the input **110** via an input device, such as a keyboard, mouse, or touchscreen. Alternatively, the processor **202** can receive the input **110** as an electronic communication via a network, such as a local area network or the Internet.

[0030] In block **304**, the processor **202** generates software-stack candidates **124***a-n* for the target software item **208**. The software-stack candidates **124***a-n* can have unique configurations of software components. For example, each software-stack candidate can have its own unique configuration of software components. The processor **202** can generate the software-stack candidates **124***a-n* by executing a search algorithm, such as a heuristic search algorithm or a stochastic search algorithm.

[0031] In block **306**, the processor **202** determines a respective score for each software-stack candidate of the software-stack candidates **124***a-n* based on the characteristic of the computing environment **210** and a unique configuration of software components forming the software-stack candidate. The processor **202** can generate the scores **126***a-n* using a predefined scoring function. Although this step is depicted as sequential to step **304**, in some examples the processor **202** can determine the respective score for each software-stack candidate as part of step **304** (e.g., as part of the search process). For example, the scoring function **128** can be used as an objective function during the search process. And each respective score may indicate the amount in which the corresponding software-stack candidate satisfies the objective function.

[0032] In block **308**, the processor **202** selects a particular software-stack candidate **124***n* from the software-stack candidates **124***a-n* as a recommended software-stack **114**. The particular software-stack candidate **124***n* can be selected based on the respective score **126***n* for the particular software-stack candidate **124***n* having a predefined attribute. For example, the processor **202** can select as the recommended software-stack **114** whichever of the software-stack candidates **124***a-n* has a maximum score, a minimum score, or a score with another predefined attribute.

[0033] In block **310**, the processor **202** generates an output **120** indicating the recommended software-stack **114**. This may enable the recommended software-stack **114** to be included (e.g., installed) in the computing environment. In some examples, the output **120** can be a display signal for outputting the recommended software-stack **114** on a display. In other examples, the output **120** can be an electronic communication transmitted over a network to a client device.

[0034] The foregoing description of certain examples, including illustrated examples, has been presented only for the purpose of illustration and description and is not intended to be exhaustive or to limit the disclosure to the